

MAgentA: Un Sistema Multi Agente per la Gestione di Agende e Riunioni

Pietro Baroni Alfonso Gerevini Paolo Toninelli
Dipartimento di Elettronica per l'Automazione
Università degli Studi di Brescia
via Branze 38, I-25123 Brescia, Italy
{baroni, gerevini}@ing.unibs.it

Abstract—In questo articolo viene presentato MAgentA, un sistema multi-agente sviluppato nel contesto del progetto interuniversitario ANEMONE allo scopo di gestire agende e pianificare riunioni tra gli utenti in modo semiautomatico. Ogni agenda è gestita da un agente che conosce gli impegni personali e le preferenze di un particolare utente, mentre le riunioni vengono organizzate da un apposito agente in grado di contattare le agende e ragionare sulle preferenze degli utenti e sulla loro disponibilità nel tempo. MAgentA, che è stato sviluppato utilizzando la piattaforma JADE e lo standard FIPA, si caratterizza rispetto ad altri sistemi analoghi poiché è in grado di discretizzare il tempo in modo guidato dalle preferenze oltre che dalla disponibilità degli utenti contattati ed utilizza criteri che cercano di ottimizzare anche la durata della riunione, entro i limiti stabiliti dall'utente che ne ha fatto richiesta.

I. INTRODUZIONE

Le tecnologie ad agenti forniscono strumenti che permettono di modellare ed implementare sistemi software sempre più articolati, in cui sono possibili astrazioni espressive come agenti, società di agenti ed ambiente [1].

In questo contesto si colloca il progetto interuniversitario ANEMONE (A Network of Multi-agent Open Environments). Il progetto si propone come obiettivo da una parte la realizzazione di una rete di piattaforme multi-agente in grado di offrire servizi dedicati alla comunità accademica ed interoperabile con altre piattaforme attive in diverse parti del mondo, e dall'altro la ricerca di soluzioni ad alcuni problemi comuni nella realizzazione di piattaforme multi-agente. Una delle funzionalità che il progetto si propone di realizzare è la gestione semiautomatica degli impegni dei docenti che fornisca un supporto all'organizzazione di riunioni. Il problema è di particolare rilevanza pratica ed è appropriato per l'applicazione della tecnologia ad agenti, come confermato dall'esistenza di svariati lavori sul tema [2], [3], [4], [5], [6].

Nell'ambito del progetto ANEMONE, tale servizio viene fornito da un sistema ad agenti denominato MAgentA (Multi Agent Agenda) e sviluppato tramite l'utilizzo di JADE (Java Agent Development Framework) [7], un framework che semplifica l'implementazione dei sistemi multi-agente attraverso un middle-ware conforme alle specifiche FIPA [8]. Alcune caratteristiche di MAgentA lo differenziano da lavori analoghi preesistenti. In particolare il metodo utilizzato per la definizione dei possibili intervalli temporali in cui pianificare una riunione. Tali intervalli vengono infatti individuati senza

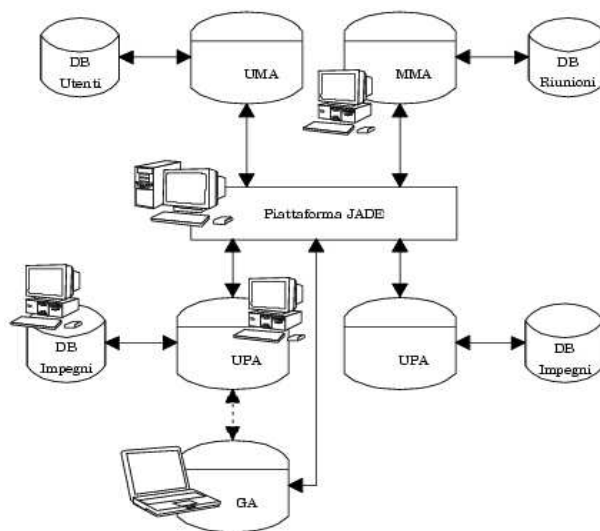


Fig. 1. Architettura del sistema. Agenti, database e connessioni.

la necessità di suddividere il tempo in intervalli discreti di dimensione fissa, tramite un algoritmo guidato dalle preferenze degli utenti ed in grado di ignorare intervalli poco rilevanti.

L'articolo descrive le funzionalità e la struttura del sistema MAgentA ed è organizzato nel modo seguente: nelle sezioni II e III verrà presentata una panoramica sull'architettura del sistema, gli agenti che lo compongono, il modo in cui essi comunicano e le funzionalità offerte; nella sezione IV si vedranno quali sono i compiti dell'agente che implementa l'agenda di un utente; nella sezione V verranno espresse le caratteristiche dell'agente incaricato della gestione degli utenti del sistema; nella sezione VI sarà esaminato l'agente che permette all'utente di interagire con la propria agenda tramite l'interfaccia grafica; nella sezione VII verranno espresse le caratteristiche dell'agente incaricato di pianificare nuove riunioni su richiesta degli utenti, tenendo in considerazione le loro preferenze. La sezione VIII conclude l'articolo sintetizzando i risultati ottenuti e menzionando i possibili sviluppi futuri di questo lavoro.

II. ARCHITETTURA E COMPONENTI DI MAGENTA

Il sistema è costituito da un insieme di agenti dotati di diverse capacità ed in grado di cooperare. Ognuno di essi

ricopre uno specifico ruolo ed è in grado di accedere ad un database in cui vengono mantenuti i dati di cui l'agente ha bisogno per assolvere ai propri compiti. I dati sono organizzati in diversi database:¹

- Un database degli utenti registrati nel sistema.
- Un database delle riunioni fissate tra gli utenti registrati.
- Un database riservato ad ogni singolo utente, in cui sono memorizzati i suoi impegni personali.

Gli agenti che costituiscono il sistema sono (figura 1):

- *User Personal Agent (UPA)*: un agente in grado di accedere al database personale dell'utente che ne è proprietario ed assumere il ruolo di partecipante durante la pianificazione di una riunione. Questo agente è in grado di recuperare tutti gli impegni dell'utente, valutare in quali intervalli possa essere considerato libero e gestire eventuali preferenze espresse dall'utente in modo da fornire all'iniziatore di una riunione le informazioni necessarie a costruire le funzioni obiettivo utilizzate per collocarla opportunamente.
- *User Management Agent (UMA)*: un agente per la gestione degli utenti che ha il compito di avviare le agende e di distribuire loro le informazioni necessarie ad effettuare l'accesso al proprio database personale. L'agente interviene inoltre durante il login degli utenti, fornendo ad ognuno l'indirizzo della propria agenda. L'UMA è il solo agente autorizzato ad accedere al database degli utenti.
- *GUI Agent (GA)*: un agente per la gestione della GUI, il cui compito fondamentale consiste nel fare da intermediario tra l'agenda e l'interfaccia utente, traducendo gli eventi della GUI in messaggi per l'UPA e viceversa.
- *Meeting Management Agent (MMA)*: un agente gestore delle riunioni a cui è consentito l'accesso al database delle riunioni ed il cui compito principale consiste nell'assumere il ruolo di iniziatore durante la pianificazione di una riunione. Questo agente è in grado di ragionare sui vincoli temporali delle riunioni e sugli impegni presenti nelle agende dei partecipanti per pianificare la nuova riunione in modo ottimale.

A sistema avviato esistono, oltre all'UMA ed al MMA, un UPA per ogni utente ed un GA per ogni utente che abbia effettuato il login sul proprio UPA.

Ogni agente "vive" all'interno di un *container* a sua volta contenuto in una piattaforma JADE (Java Agent Development Framework) [10] e provvede a registrarsi presso il DF (Directory Facilitator conforme alle specifiche FIPA) [8] messo a disposizione della piattaforma.² Tale agente permette di ottenere informazioni circa gli agenti registrati e viene utilizzato per recuperare l'indirizzo di UMA e MMA ogni volta che un'agenda abbia bisogno di contattare uno di questi agenti.

¹I database possono essere distribuiti su più server. La connessione è stata realizzata tramite JDBC, mentre per la permanenza dei dati si è utilizzato JPOX[9], un'implementazione dello standard JDO. In questo modo sia il formato dei database che il tipo di server possono essere configurati.

²Ogni container può essere lanciato su una diversa macchina in modo da distribuire gli agenti che compongono il sistema. Il corretto indirizzamento degli agenti viene gestito tramite il DF.

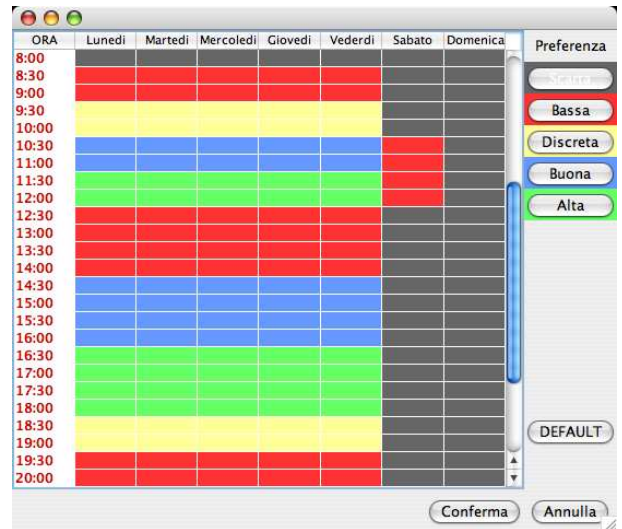


Fig. 2. Configurazione delle preferenze di un utente.

Ogni volta che viene iniziata una conversazione tra gli agenti, l'iniziatore genera un identificativo (*Conversation-ID*) univoco da assegnare a tutti i messaggi che verranno scambiati durante la conversazione. In questo modo ogni agente può essere impegnato in più conversazioni senza che queste interferiscano tra loro.

III. FUNZIONALITÀ

Nella versione attuale il sistema offre alcune funzionalità che permettono ad un utente di configurare la propria agenda esprimendo delle preferenze circa eventuali riunioni a cui dovrà prendere parte, gestire i propri impegni personali ed interagire con altre agende allo scopo di pianificare una riunione. Più in dettaglio, le funzionalità attualmente disponibili sono:

- Configurazione delle preferenze: ogni utente ha la possibilità di esprimere un *grado di disponibilità* (preferenza) a partecipare ad eventuali riunioni in un determinato intervallo di tempo all'interno di una settimana tipo. Tali preferenze possono essere espresse con una granularità di mezz'ora tramite l'utilizzo di cinque livelli di disponibilità: basso, discreto, buono, medio e scartato (figura 2). L'ultimo livello viene considerato come un impegno fittizio che impedisce all'utente di partecipare a qualsiasi riunione che vi si sovrapponga. Gli altri livelli corrispondono invece a dei valori numerici che quantificano le preferenze dell'utente.
- Inserimento e modifica di impegni personali: ogni utente può agire sul proprio database personale tramite un'interfaccia che permette di aggiungere, rimuovere e modificare gli impegni traducendo ogni azione in opportuni messaggi da inviare all'UPA ed aggiornando quanto presentato a video all'utente.
- Richieste di riunione: ogni utente può assumere il ruolo di iniziatore definendo dei vincoli per una nuova riunione

ed inviando una richiesta al *MMA*, il quale interagirà con le agende degli utenti invitati a prendervi parte.

- Ricerca e ordinamento soluzioni: l'agente gestore delle riunioni è in grado di valutare le possibili collocazioni temporali individuate per una nuova riunione allo scopo di determinarne la qualità in relazione alle preferenze espresse dalle agende degli utenti invitati. Ciò permette di ordinare le collocazioni individuate in modo da identificare quelle che meglio soddisfano la richiesta.
- Selezione di una soluzione, memorizzazione e notifica agli utenti: gli intervalli di tempo selezionati e valutati possono essere analizzati dall'iniziatore allo scopo di verificarne la qualità. L'utente potrà quindi selezionarne uno da assegnare alla riunione che si vuole fissare ed inviare una notifica alle agende degli utenti invitati.

IV. UPA: AGENTE PERSONALE DI UN UTENTE

L'*UPA* rappresenta l'agenda vera e propria di un utente. Questo agente è in grado di accedere al database personale del suo proprietario per inserirvi nuovi impegni ed estrarre o modificare quelli esistenti. I compiti principali di questo agente sono:

- rispondere alle richieste provenienti dal *GA*, in modo da permettere all'utente di visualizzare ed interagire con la propria agenda;
- interagire con il *MMA* assumendo il ruolo di partecipante durante le conversazioni effettuate tramite il protocollo FIPA-Contract-Net, allo scopo di fornire all'iniziatore gli intervalli di tempo disponibili per pianificare una nuova riunione in accordo con gli impegni del proprio utente;
- valutare tali intervalli tenendo in considerazione gli impegni personali, le riunioni e le loro eventuali ricorrenze;
- valutare le eventuali preferenze dell'utente traducendole in un opportuno insieme di valori numerici.

Nel momento in cui l'*UPA* viene contattato dal *MMA* allo scopo di fissare una nuova riunione, esso si attiva valutando le possibilità e le preferenze del suo proprietario. Le possibilità dell'utente vengono definite analizzando la lista di tutti i suoi impegni ed estraendo tutti gli intervalli in cui l'utente risulta libero. Per quanto riguarda le preferenze l'agente analizza la configurazione effettuata dall'utente su di una settimana tipo e la applica alle possibilità dell'utente. In figura 2 è riportata la finestra che permette all'utente di configurare le proprie preferenze.

V. UMA: AGENTE GESTORE DEGLI UTENTI

Il primo compito dello *UMA* consiste nell'avviare le agende, prelevando le informazioni necessarie dal database degli utenti registrati. Per ogni utente sono definite le seguenti informazioni:

- il nome dell'utente, che deve essere unico in quanto verrà utilizzato come identificativo;
- il nome dell'agenda di cui l'utente è proprietario, ovvero il nome da assegnare allo *UPA* di quel particolare utente;
- il nome del database di cui l'utente è proprietario;

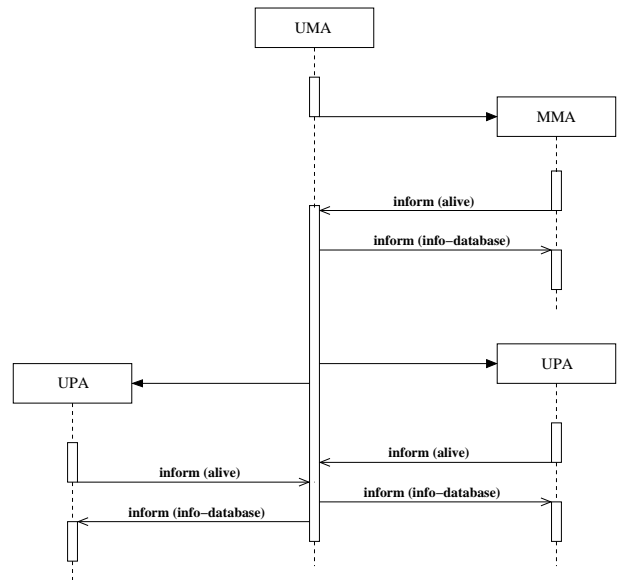


Fig. 3. Scambio di messaggi all'avvio delle agende.

- il nome utente e la password che l'agenda in questione dovrà utilizzare per poter accedere al proprio database personale;
- la categoria a cui l'utente appartiene

Il meccanismo di avvio delle agende coinvolge tutti gli agenti (tranne il *GA*, che esiste solo nel momento in cui un utente effettua un login sulla propria agenda) ed è costituito dai seguenti passi (figura 3):

- 1) lo *UMA* recupera dal proprio database la lista degli utenti, avvia un *MMA* e si mette in attesa di un suo messaggio di conferma;
- 2) ricevuta una conferma (*inform (alive)*) dal *MMA* lo *UMA* avvia tutte le agende (*UPA*) di cui gli utenti prelevati dal database sono proprietari e attende conferma da ognuna;
- 3) ricevuto un *inform (alive)* da parte di un'agenda, lo *UMA* invia allo *UPA* il nome del proprio database personale ed il nome utente e la password che ne consentono l'accesso;

Durante il meccanismo di avvio vengono effettuati dei controlli per impedire che il sistema si trovi in situazioni anomale. Tali controlli impediscono ad esempio che un'agenda venga avviata qualora nel sistema non vi sia nessun agente che offra il servizio di gestione delle riunioni.

Una volta avviate le agende, ogni *UPA* si pone in uno stato di attesa per eventuali messaggi provenienti dal *MMA* o da un eventuale *GA*, che può cercare di contattare un'agenda per effettuare il login.

VI. GA: AGENTE GESTORE DELLA GUI

Il *GA* deve essere lanciato dall'utente che vuole effettuare un login sulla propria agenda. Questo agente avvia un'interfaccia grafica tramite la quale l'utente può interagire con il proprio

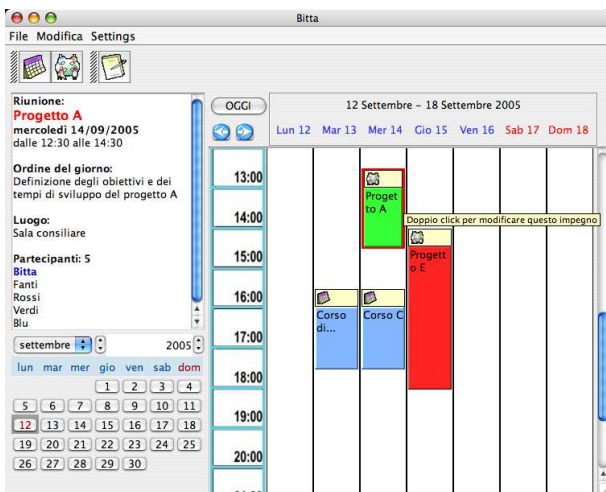


Fig. 4. Interfaccia presentata all'utente dal GUI Agent.

UPA, inserendo o modificando degli impegni. I compiti di tale agente sono:

- 1) permettere il login su di un UPA, contattando lo UMA allo scopo di identificare l'utente e l'agenda di cui egli è proprietario;
- 2) fornire un servizio di traduzione delle azioni eseguite dall'utente in opportuni messaggi da inviare allo UPA affinché l'azione richiesta venga eseguita;
- 3) fornire un servizio di traduzione dei messaggi provenienti dallo UPA in opportune azioni dell'interfaccia che permettano di visualizzare le informazioni contenute nell'agenda;

In figura 4 è mostrata l'interfaccia che viene presentata all'utente dopo che è stato effettuato il login.

A. Login/logout di un utente

Una volta avviato, il GA provvede a registrarsi presso il DF e richiede all'utente di immettere nome e password per effettuare il login. Inserite queste informazioni vengono effettuati i passi seguenti (figura 5):

- 1) il GA contatta il gestore degli utenti, UMA, inviando il nome utente e chiedendo che gli venga fornito l'indirizzo dell'agenda di cui l'utente è proprietario
- 2) lo UMA identifica l'utente ed il suo UPA, cerca tale agente tra quelli registrati nella piattaforma ed invia il suo indirizzo al GA; nel caso in cui non esista un'agenda per l'utente specificato, il gestore degli utenti risponde con un failure che termina il protocollo;
- 3) il GA contatta lo UPA inviando una richiesta di login. Se l'utente risulta già loggato l'UPA rifiuta la richiesta, altrimenti chiede al GA la password di accesso e verifica l'identità dell'utente confrontando nome e password con le informazioni memorizzate nel proprio database personale;
- 4) se il confronto ha successo il login viene accettato, viene registrato l'indirizzo del GA che lo ha effettuato il login e a quest'ultimo viene inviato un messaggio di conferma

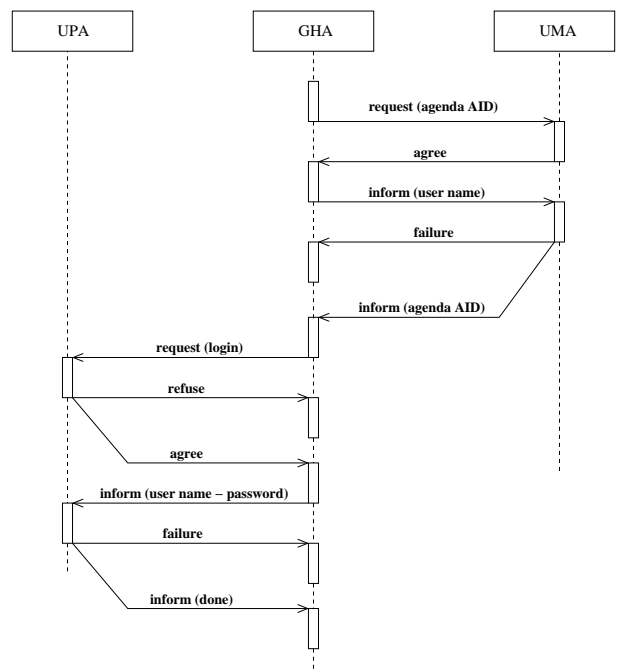


Fig. 5. Messaggi scambiati dagli agenti durante il meccanismo di login di un utente.

(inform (done)), in caso contrario viene inviato un messaggio di fallimento (failure);

Al termine del meccanismo di login lo UPA conosce l'indirizzo del GA e si pone in attesa di eventuali richieste provenienti da questo agente.

Il meccanismo di logout viene invece effettuato senza che venga coinvolto il gestore degli utenti. In questo caso il GA invia la richiesta di logout direttamente all' UPA, il quale deregistra tale agente e si pone in uno stato in cui non possono essere accettate richieste provenienti dal GA.

B. Interazione con l'agenda

Dopo avere effettuato il login, il sistema permette agli utenti di inserire, rimuovere e modificare impegni personali all'interno della propria agenda. Tutte le funzionalità vengono realizzate inviando dei messaggi di richiesta allo UPA di proprietà dell'utente.

Le conversazioni tra il GA e l'UPA vengono effettuate utilizzando una versione modificata del protocollo FIPA-Request. Il protocollo utilizzato prevede un inform aggiuntivo la cui utilità consiste nel poter inviare al destinatario delle informazioni necessarie a portare a termine la richiesta una volta che questa sia stata accettata.³ Il protocollo utilizzato è riportato in figura 6:

- 1) il GA invia allo UPA una richiesta in cui viene specificato il tipo di azione che si vuole venga effettuata;

³Ad esempio una tipica richiesta della GUI verso l'agenda è costituita da un messaggio request che contiene l'azione che si vuole venga eseguita (es. richiesta di invio degli impegni), mentre tutti i parametri necessari ad eseguirla (es. periodo di tempo di cui si vogliono gli impegni) vengono forniti nell'inform successivo.

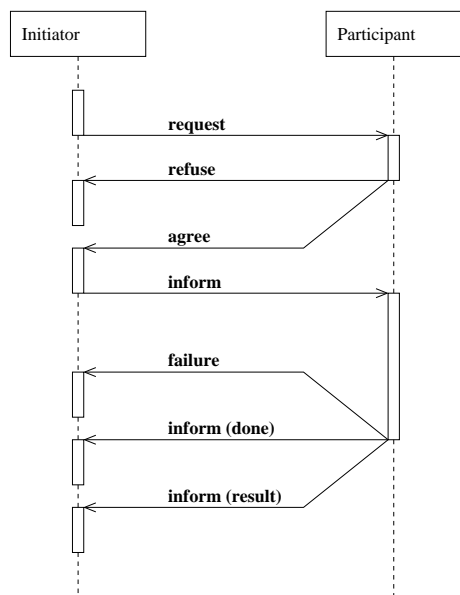


Fig. 6. Protocollo FIPA-request con l'aggiunta di un messaggio inform

- 2) lo *UPA* decide se accettare o rifiutare la richiesta ed invia un messaggio di conferma o di rifiuto al *GA*;
- 3) se la richiesta viene accettata, il *GA* invia un *inform* contenente eventuali parametri necessari a soddisfare la richiesta;
- 4) lo *UPA* esegue l'operazione richiesta: in caso di successo viene inviato al *GA* un *inform* contenente il risultato dell'operazione, in caso contrario viene inviato un messaggio di fallimento (*failure*) che conclude la conversazione.

L'interazione con il gestore della GUI da parte dell'agenda viene effettuata da un *behaviour* strutturato come una macchina a stati finiti costituita dai seguenti stati (figura 7):

- 1) stato di accettazione delle richieste: l'agenda accetta qualsiasi messaggio di richiesta proveniente dal gestore della GUI che ha effettuato il login, ricevuto un messaggio l'agenda si porta nello stato successivo;
- 2) stato di identificazione della richiesta: il messaggio ricevuto viene utilizzato per identificare l'operazione richiesta; in questo stato l'agenda decide se accettare o meno la richiesta ed invia un messaggio di conferma o di rifiuto al *GA*;
- 3) uno stato di rifiuto in cui l'operazione richiesta non viene eseguita e la conversazione viene terminata inviando un *refuse* al *GA*;
- 4) uno stato per ogni possibile operazione: una volta identificata ed accettata la richiesta l'agenda si pone in uno dei possibili stati, in attesa di eventuali parametri da parte del *GA*; una volta ricevuto il messaggio *inform* contenente tali parametri l'operazione viene portata a termine;
- 5) stato conclusivo: vengono inviati al *GA* i risultati dell'operazione (oppure un messaggio di fallimento se essa

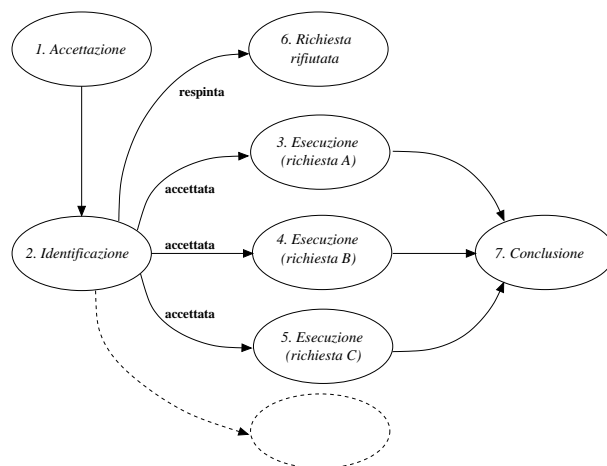


Fig. 7. Macchina a stati finiti che implementa l'interazione con la GUI da parte dell'agenda. Lo stato tratteggiato rappresenta l'esecuzione di ulteriori possibili richieste.

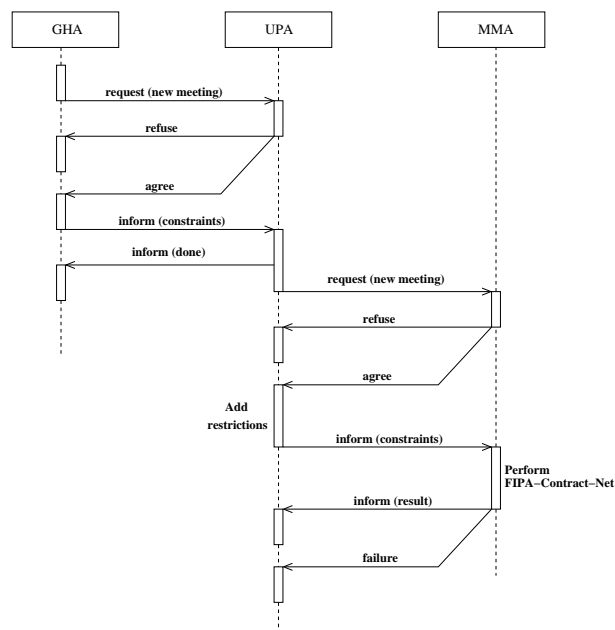


Fig. 8. Messaggi scambiati durante la richiesta di una nuova riunione da parte di un utente.

non è andata a buon fine) e si riporta la macchina allo stato di accettazione;

VII. MMA: GESTIONE DELLE RIUNIONI

Il *MMA* è costantemente in attesa di richieste provenienti dagli *UPA* ed è in grado di fornire due servizi:

- 1) inviare ad un *UPA*, qualora ne faccia richiesta, le riunioni a cui l'utente che ne è proprietario deve partecipare (perchè possano essere mostrate all'utente oppure per definire gli intervalli in cui egli è libero da impegni);
- 2) assumere il ruolo di organizzatore di nuove riunioni, in seguito ad una richiesta di nuova riunione da parte di un *UPA*.

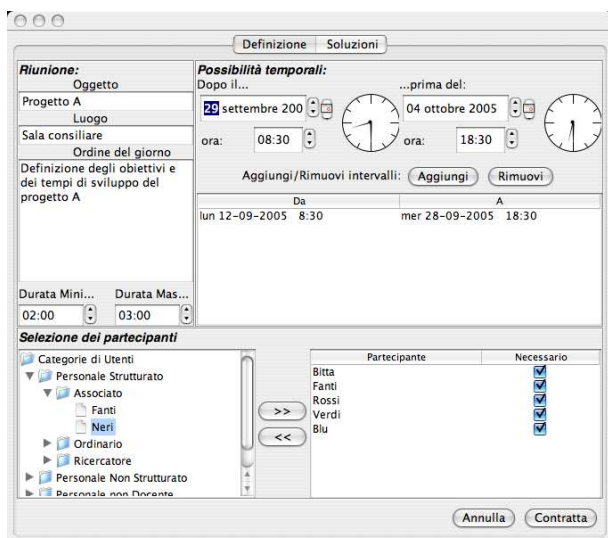


Fig. 9. Inserimento dei vincoli per una nuova riunione.

Ogni utente può richiedere che venga fissata una riunione (figura 9) e definire dei vincoli temporali sulla sua collocazione. La richiesta dell'utente viene inviata dal *GA* allo *UPA* che provvede ad attivare il *MMA*, il quale si assumerà l'incarico di contattare le agende di tutti gli utenti invitati a prendere parte alla riunione e di elaborare delle possibili soluzioni che rispettino i vincoli temporali che sono stati imposti. I messaggi scambiati dagli agenti in questa fase sono riportati in figura 8. La richiesta di fissare una nuova riunione può essere rifiutata dal *MMA*.⁴ La conversazione tra il *GA* e lo *UPA* viene sospesa non appena lo *UPA* abbia ricevuto i vincoli definiti dall'utente, in attesa che il *MMA* elabori delle possibili soluzioni. Durante la conversazione con il *MMA* lo *UPA* provvede ad aggiungere altri vincoli (determinati dagli impegni dell'utente che ha iniziato la richiesta). Spetta invece al *MMA* il compito di soddisfare la richiesta. Per fissare una nuova riunione l'utente deve fornire:

- la descrizione della riunione da pianificare, ovvero l'oggetto della riunione, il luogo in cui si dovrà tenere e opzionalmente una descrizione degli argomenti da trattare;
- uno o più intervalli temporali alternativi entro i quali vincolare la collocazione della riunione;
- la durata minima ed eventualmente quella massima da assegnare alla riunione. Nel caso in cui la durata massima non venga fornita essa verrà posta uguale a quella minima e si cercherà di fissare la riunione esattamente con quella durata;
- gli utenti invitati a partecipare. Essi possono essere selezionati da una struttura ad albero in cui sono mostrati tutti gli utenti registrati, organizzati in categorie. Inoltre l'iniziatore può marcare alcuni utenti come *non neces-*

⁴Attualmente, ad esempio, essa viene rifiutata nel caso in cui essa coinvolga utenti che sono impegnati a fissare un'altra riunione, poiché il sistema non è in grado di gestire in modo corretto richieste concorrenti. L'inserimento di tale funzionalità potrà essere valutato in futuro.

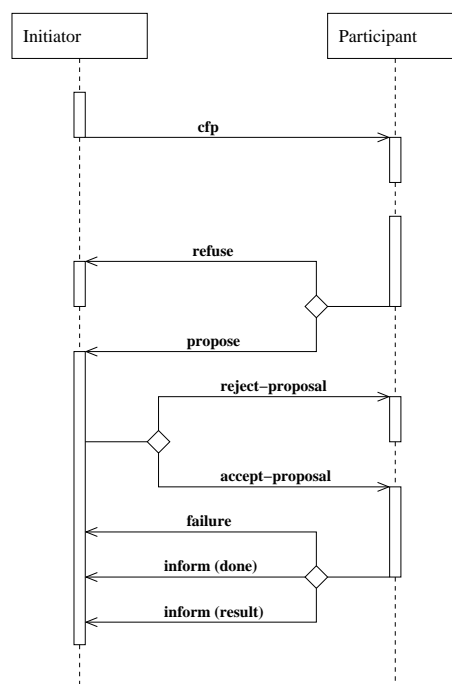


Fig. 10. Protocollo FIPA-Contract-Net

sari, permettendo al sistema di tenere conto di questa indicazione se ciò può essere di aiuto all'identificazione di possibili soluzioni.

A questi vincoli si aggiunge quello costituito dalla *non sovrapposizione* con eventuali impegni dell'utente iniziatore. Tali informazioni vengono inviate al *MMA*, il quale inizia una conversazione con le agende degli utenti invitati utilizzando il protocollo FIPA-Contract-Net (figura 10):

- 1) il *MMA* invia allo *UPA* di ogni utente invitato un *cfp* contenente tutte le informazioni circa la riunione che si vuole fissare;
- 2) se l'*UPA* è già impegnato a fissare un'altra riunione risponde al *MMA* con un *refuse* allo scopo di evitare il caso di richieste concorrenti. Altrimenti l'*UPA* esamina i propri impegni e costruisce una lista di intervalli in cui il proprio utente può essere considerato libero; questa lista (eventualmente vuota)⁵ viene inviata al *MMA* tramite un *propose*, assieme alle preferenze che l'utente ha espresso configurando la propria agenda;
- 3) il *MMA* esamina le risposte ricevute estraendo una lista di possibili collocazioni per la riunione;
- 4) se tale lista è vuota il *MMA* invia ad ogni *UPA* contattato un messaggio *reject-proposal* che conclude la conversazione; in caso contrario viene inviato un *accept-proposal*;
- 5) lo *UPA* invia un *inform* di conferma che conclude la conversazione.

⁵il *MMA* è in grado di identificare anche collocazioni temporali in cui non tutti gli utenti sono disponibili e di valutare la qualità di tali collocazioni in base alla percentuale di utenti che possono effettivamente partecipare alla riunione.

Al passo 3 il *MMA* analizza le risposte ricevute dalle agende, da cui estrae delle *possibili soluzioni* per la richiesta di riunione. Tali soluzioni sono definite come segue:

Definizione 7.1: Una **possibile soluzione** per una richiesta di riunione è un qualsiasi intervallo di tempo che soddisfi tutti i vincoli temporali definiti dall'iniziatore (compreso quello di non sovrapposizione con gli impegni di tale utente).⁶ Nelle prossime sezioni verrà illustrato il modo in cui l'*MMA* identifica le possibili e soluzioni e ne valuta la qualità.

A. Funzioni obiettivo e valutazione

Prima di identificare le possibili soluzioni il gestore delle riunioni esamina le risposte delle agende invitate e costruisce le *funzioni obiettivo* che permetteranno di valutare la qualità degli intervalli individuati. Tali funzioni vengono utilizzate anche per la selezione di un sottoinsieme delle possibili soluzioni, allo scopo di limitarne il numero (che altrimenti potrebbe essere infinito) ed evitare di considerare soluzioni di scarsa rilevanza.

Per ogni utente i invitato alla riunione vengono costruite, sulla base della risposta ricevuta dal suo *UPA*, due funzioni, che chiamiamo *disponibilità* ($d_i(t)$) e *preferenza* ($p_i(t)$) dell'utente i -esimo, dove t rappresenta il tempo.

Definizione 7.2: La **funzione di disponibilità** $d_i(t)$ dell' i -esimo utente assume i seguenti valori

$$d_i(t) = \begin{cases} 1 & \text{utente non impegnato al tempo } t \\ 0 & \text{utente impegnato al tempo } t \end{cases}$$

La funzione di preferenza $p_i(t)$ viene costruita a partire dalle preferenze espresse dall' i -esimo utente su di una settimana tipo. Esse vengono tradotte in un insieme di *valori di preferenza*, ovvero valori discreti all'interno di un range predefinito.

Definizione 7.3: La **funzione di preferenza** $p_i(t)$ assume il valore di preferenza espresso dall'utente i -esimo per il giorno della settimana e l'ora in cui ricade l'istante t .

A partire dalle funzioni di disponibilità e preferenza degli utenti invitati alla riunione, vengono costruite due funzioni obiettivo: la **disponibilità media** $d(t)$ e la **preferenza media** $p(t)$ dei partecipanti alla riunione. La disponibilità media $d(t)$ degli utenti invitati è definita nel seguente modo:

$$d(t) = \frac{1}{n} \sum_{i=1}^n d_i(t) \quad (1)$$

mentre $p(t)$ è definita come:

$$p(t) = \frac{1}{n} \sum_{i=1}^n p_i(t) \quad (2)$$

dove n rappresenta il numero di utenti invitati a partecipare alla riunione.

Le possibili soluzioni identificate per soddisfare la richiesta di riunione, costituite da intervalli temporali per la sua collocazione, vengono valutate in base alle funzioni obiettivo che

⁶Sovrapposizioni con impegni di altri utenti sono in generale permesse, tuttavia verranno in seguito esaminate ed utilizzate allo scopo di valutare la qualità degli intervalli individuati.

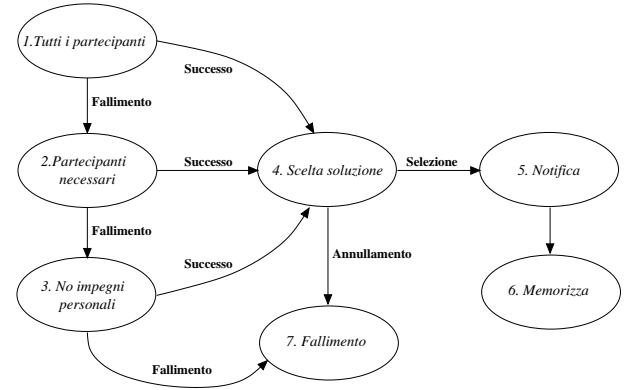


Fig. 11. Macchina a stati finiti per l'identificazione degli intervalli da esaminare per la ricerca delle soluzioni rilevanti.

sono state definite. Ad ogni soluzione vengono associati due indici α e β ottenuti integrando le funzioni obiettivo all'interno dell'intervallo da valutare. In particolare, data una soluzione $\omega = [t_0, t_1]$, si ha

$$\alpha_\omega = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} d(t) \delta t \quad (3)$$

$$\beta_\omega = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} p(t) \delta t \quad (4)$$

dove la normalizzazione viene effettuata per rendere confrontabili intervalli di diversa durata.¹

Le soluzioni individuate vengono quindi ordinate sulla base degli indici α e β e della loro durata. Il criterio di ordinamento tra due intervalli $\omega_1 = [t_0, t_1]$ e $\omega_2 = [t_2, t_3]$ è il seguente:

$$\omega_1 \prec \omega_2 \begin{cases} \alpha_1 > \alpha_2 \vee \\ \alpha_1 = \alpha_2 \wedge \beta_1 > \beta_2 \vee \\ \alpha_1 = \alpha_2 \wedge \beta_1 = \beta_2 \wedge D_1 > D_2 \vee \\ \alpha_1 = \alpha_2 \wedge \beta_1 = \beta_2 \wedge D_1 = D_2 \wedge t_0 < t_2 \end{cases} \quad (5)$$

dove con D_1 e D_2 sono state indicate rispettivamente la durata dell'intervallo ω_1 e quella di ω_2 . Si procede confrontando ordinatamente α , β ed infine la durata delle soluzioni in esame. Nel caso in cui questi parametri risultino tutti coincidenti si rispetta l'ordinamento temporale definito dall'istante iniziale di ogni soluzione.

B. Individuazione delle possibili soluzioni

Le soluzioni, ovvero gli intervalli in grado di soddisfare alla richiesta, vengono individuate tramite una macchina a stati finiti (riportata in figura 11) che attualmente è costituita da tre fasi di ricerca di soluzioni:

¹Se gli indici α e β non venissero normalizzati rispetto alla durata di ω , intervalli molto lunghi ma di scarsa qualità rispetto alle funzioni obiettivo potrebbero essere privilegiati rispetto a intervalli più brevi ma migliori dal punto di vista delle funzioni di disponibilità e preferenza.

- 1) si tengono in considerazione gli impegni di tutti gli utenti invitati: viene costruita una lista costituita dagli intervalli temporali in cui tutti gli utenti invitati risultano liberi da impegni;
- 2) in caso di fallimento si passa a considerare solo gli impegni di tutti gli utenti invitati come partecipanti *necessari*: viene costruita una lista costituita da intervalli temporali in cui tutti gli invitati necessari risultano liberi da impegni;
- 3) se anche lo stadio 2 si conclude con un fallimento si cerca una soluzione considerando solo gli impegni dell'iniziatore e le riunioni che coinvolgono i partecipanti necessari: viene costruita una lista costituita dagli intervalli temporali in cui l'iniziatore risulta libero da qualsiasi impegno e gli invitati necessari risultano liberi da altre riunioni in cui la loro presenza sia necessaria.
- 4) nel caso in cui non si riesca a definire alcun possibile intervallo in cui pianificare la riunione, il procedimento termina avvertendo del fallimento l'iniziatore, il quale potrà eventualmente modificare la richiesta ridefinendone i vincoli temporali.

Nel caso in cui tutte le fasi falliscano la macchina termina in uno stato di fallimento e la richiesta non viene soddisfatta. In caso contrario, la lista L degli intervalli che sono stati individuati e le funzioni obiettivo $d(t)$ e $p(t)$ vengono utilizzati per costruire un insieme di *istanti rilevanti* da utilizzare per identificare le possibili soluzioni.

Definizione 7.4: Chiamiamo T l'insieme degli istanti rilevanti. Un istante t_k è rilevante se

- t_k coincide con l'istante iniziale o finale di uno degli intervalli contenuti in L ,
- t_k coincide con un punto di salto di almeno una delle funzioni obiettivo $d(t)$ e $p(t)$.⁷

La lista di intervalli L e l'insieme T degli istanti rilevanti, vengono analizzati dall'*Algoritmo di Identificazione delle Soluzioni Rilevanti* (figura 12) allo scopo di estrarne un insieme di possibili soluzioni. Tale algoritmo rappresenta un elemento di novità rispetto alle soluzioni proposte in [2], [3], [5], [6]. Anzichè discretizzare il tempo in intervalli predefiniti l'algoritmo individua degli intervalli che sono rilevanti al fine di ottimizzare la collocazione di una riunione rispetto alle funzioni obiettivo. Vengono considerati tutti e soli gli intervalli temporali che sono candidati a rappresentare dei massimi locali per gli indici α e β . Questo permette di ottenere un numero finito di possibili soluzioni con il vantaggio di una maggiore flessibilità.

Esempio: Nella situazione riportata in figura 13 si ha un intervallo da analizzare e tre istanti temporali rilevanti t_0 , t_1 e t_2 , di cui t_0 e t_2 rappresentano l'istante iniziale e finale dell'intervallo da analizzare, mentre t_1 rappresenta un punto di salto di una delle funzioni obiettivo. L'algoritmo di individuazione delle soluzioni inizia individuando una soluzione di lunghezza minima posizionata a partire dall'istante t_0 . Successivamente

Algoritmo di Identificazione delle Soluzioni Rilevanti

input: un insieme L di intervalli da esaminare, la durata minima e massima d_{min} e d_{max} , un insieme di *istanti rilevanti* T
output: Un insieme R di soluzioni.

```

1.  $R \leftarrow \emptyset$ 
2. for each  $I \in L$  do
3.    $\omega = [I_{start}, I_{start} + d_{min}]$ 
4.   while  $\omega_{end} < I_{end}$  do
5.      $R = R + \omega$ 
6.     if  $(\omega_{end} - \omega_{start}) < d_{max}$  then
7.        $\omega_{end} = \min \{next(T, \omega_{end}), (\omega_{start} + d_{max})\}$ 
8.     else
9.       if  $(next(T, \omega_{end}) - \omega_{end}) <$ 
10.         $(next(T, \omega_{start}) - \omega_{start})$  then
11.          $\omega_{start} = next(T, \omega_{end}) - d_{max}$ 
12.       else
13.          $\omega_{start} = next(T, \omega_{start})$ 
14.          $\omega_{end} = \omega_{start} + d_{min}$ 
15.   return  $R$ 

```

Fig. 12. Algoritmo per l'identificazione delle soluzioni. I pedici *start* e *end* indicano rispettivamente l'istante iniziale e finale di un intervallo. La funzione $next(T, t)$ restituisce il primo istante temporale successivo a t tra quelli contenuti in T .

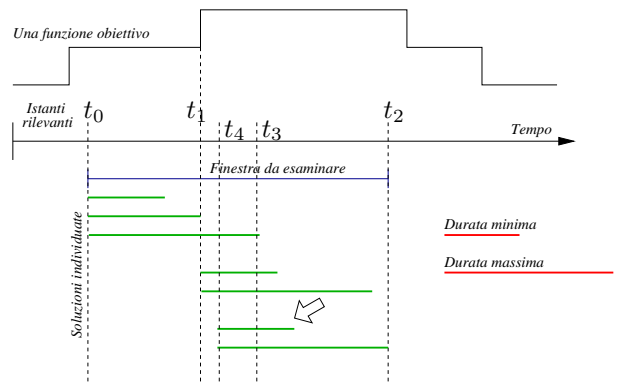


Fig. 13. Esempio di individuazione delle soluzioni rilevanti all'interno di uno dei possibili intervalli da esaminare. La soluzione indicata dalla freccia è quella individuata al passo 10 dell'algoritmo.

l'algoritmo espande la durata, individuando una soluzione che inizia in t_0 e termina in t_1 . Cercando nuovamente di espandere la durata della soluzione non è possibile individuare altri istanti rilevanti che precedano t_3 , il quale rappresenta l'istante finale della soluzione a durata massima con inizio in t_0 . Dopo aver memorizzato la soluzione compresa tra t_0 e t_3 si procede cercando di spostare l'istante iniziale. Esso viene posizionato in t_1 , e l'algoritmo riparte considerando la soluzione di durata minima e cercando di espanderla sino alla durata massima. Infine l'istante iniziale viene posizionato in t_4 , calcolato in modo da poter espandere la durata della soluzione sino al massimo valore possibile restando nei limiti imposti dall'intervallo analizzato (la soluzione a durata massima con istante iniziale in t_4 termina in t_2 , ultimo istante utile nell'intervallo a cui ci si è vincolati). Tutte le soluzioni individuate vengono valutate, ordinate e comunicate all'agente *GA* dell'iniziatore, che le presenta all'utente (figura 14) il quale ne può selezionare

⁷Tali funzioni sono discontinue a tratti per costruzione

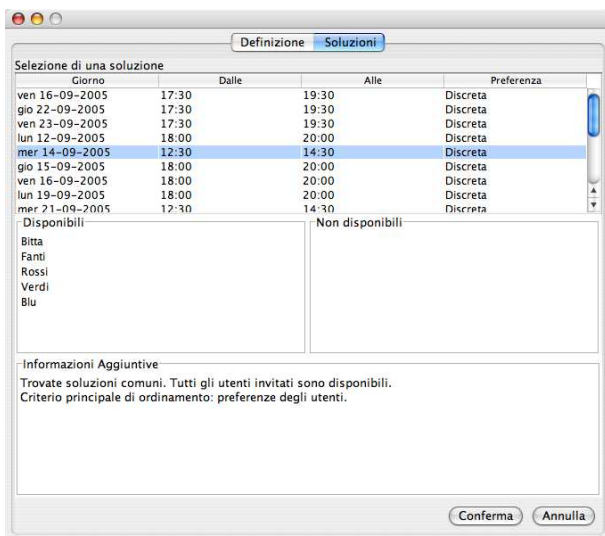


Fig. 14. Possibili soluzioni presentate all'utente iniziatore.

una da assegnare definitivamente alla nuova riunione. Tale scelta potrà essere effettuata esaminando le caratteristiche di ciascuna soluzione (qualità stimata, massimo numero di utenti non disponibili, identità di tali utenti e percentuale di utenti disponibili in media). L'intervallo selezionato sarà quindi assegnato alla riunione che potrà essere notificata agli utenti invitati e memorizzata nel database.⁸

VIII. CONCLUSIONI E SVILUPPI FUTURI

In questo articolo abbiamo descritto MAgentA, un sistema multi agente che offre un servizio di gestione degli impegni degli utenti con la possibilità di pianificare nuove riunioni in modo semiautomatico. Ogni utente può inserire e modificare a piacimento i propri impegni personali ed inviare al sistema richieste per pianificare nuove riunioni. Tali richieste sono gestite trovando collocazioni temporali per una riunione che soddisfino le disponibilità degli utenti invitati, o almeno di un loro sottoinsieme, secondo alcuni criteri che cercano di minimizzare i conflitti con altri impegni e riunioni, oltre che di massimizzare il soddisfacimento delle preferenze dei partecipanti. La bontà delle soluzioni ottenute viene stimata in base al numero effettivo di utenti disponibili in ogni possibile soluzione ed alle preferenze espresse dagli utenti.

Il sistema MAgentA presenta alcuni aspetti che lo differenziano dai lavori preesistenti sulla stessa tematica (come ad esempio [2], [3], [5], [6]). In particolare:

- 1) MAgentA utilizza lo standard FIPA [8] per la comunicazione tra gli agenti: le conversazioni sono state realizzate utilizzando messaggi conformi allo standard FIPA-ACL [11], mentre per l'identificazione di possibili

intervalli in cui posizionare una riunione è stato utilizzato il protocollo FIPA-Contract-Net; questo rende il sistema aperto all'integrazione con piattaforme conformi allo standard;

- 2) nella pianificazione di una riunione il tempo non viene discretizzato in intervalli fissi e noti a priori, ma suddiviso in intervalli dipendenti dall'andamento di funzioni obiettivo costruite per soddisfare una particolare richiesta;
- 3) le possibili collocazioni temporali di una riunione vengono identificate da un algoritmo guidato da funzioni obiettivo, capace di considerare soluzioni di diversa durata e di scartare le possibilità di scarsa rilevanza.

Il sistema è stato valutato sperimentalmente costruendo un database di agende e posizionando gli impegni di ciascuna in modo tale da verificare il comportamento del software in diverse situazioni. I risultati ottenuti in questa analisi sperimentale preliminare hanno mostrato la capacità di MAgentA di pianificare riunioni in modo efficiente rispetto alle possibilità degli utenti ed alle loro preferenze. In futuro si prevede di effettuare una fase di sperimentazione più approfondita che permetta di verificare il comportamento, le prestazioni ed i limiti del sistema proposto all'interno di un contesto realistico.

Alcuni possibili sviluppi futuri del sistema sono: il raffinamento dei vincoli (es. controllo sul numero legale di partecipanti prima di memorizzare la riunione); la possibilità di esprimere vincoli di ordinamento rispetto ad altre riunioni; la gestione di richieste concorrenti e di spostamento di riunione; lo sviluppo di un'interfaccia WEB; l'integrazione con tecnologie standard per la gestione di calendari.

REFERENCES

- [1] N.R.Jennings and M.Wooldridge, *Applications of Intelligent Agents*. University of London: Queen Mary and Westfield College, 1998, in *Agent Technology: Foundations, Applications and Markets*, 3-28.
- [2] X. D. Ahlem Ben Hassine and T. B. Ho, *Agent Based Approach to Dynamic Meeting Scheduling Problems*. AAMAS'04, 2004, vol. 3, pp.1132-1139.
- [3] E. Crawford and M. Veloso, *Mechanism Design for Multi-Agent Meeting Scheduling Including Time Preferences, Availability, and Value of Presence*. Beijing, China: in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, September 2004.
- [4] —, "Learning Dynamic Time Preferences in Multi-Agent Meeting Scheduling," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, Tech. Rep., 2005.
- [5] N.R.Jennings and A.J.Jackson, *Agent Based Meeting Scheduling: A Design and Implementation*. Electronics Letters, The Institution of Electrical Engineering, 31 (5), March 1995, pp. 350-352.
- [6] S. Sen, *An automated distributed meeting scheduler*. *IEEE Expert*, July/August 1997, vol. 12, no. 4, pp. 41-45.
- [7] *JADE web site*. <http://jade.cselt.it/>.
- [8] *FIPA web site*. <http://www.fi.pa.org>.
- [9] *JPOX web site*. <http://www.jpox.org>.
- [10] F. B. et al., *JADE Administrator's guide*, January 2005, <http://jade.cselt.it/doc/index.html>.
- [11] *FIPA ACL specifications*. <http://www.fi.pa.org/repository/aclspecs.html>.

⁸Gli utenti che si trovano "on line" in quel momento riceveranno un messaggio di notifica che potranno confermare o rifiutare. In ogni caso la riunione verrà memorizzata, ma in caso di rifiuto l'iniziatore verrà avvertito dell'assenza dell'utente. Gli utenti "off line" accettano per default tutte le notifiche.